Exam

Duration: 3 hours. All paper documents permitted. The numbers $[n]$ in the margin next to questions are indications of duration and difficulty, not necessarily of the number of points you might earn from them. You must justify all your answers.

Exercise 1 (First-Order Logic with Transitive Reflexive Relations)**.** We consider the first-order logic $\mathrm{FO}(\downarrow^*, (P_a)_{a\in\Sigma})$ over finite unranked trees labelled by some finite alphabet Σ along with the descendant relation $\downarrow^*$.

[1] 1. Give a closed first-order formula ψ_1 enforcing that the sequence of labels along any branch is in $(ab)^+$. *Hint: You can use the following first-order formulæ:*

$$x \downarrow^+ y \overset{def}{=} x \downarrow^* y \wedge x \neq y \;, \qquad x \downarrow y \overset{def}{=} x \downarrow^+ y \wedge \neg\exists z(x \downarrow^+ z \wedge z \downarrow^+ y) \;,$$
$$\mathsf{root}(x) \overset{def}{=} \neg\exists y(y \downarrow^+ x) \;, \qquad \mathsf{leaf}(x) \overset{def}{=} \neg\exists y(x \downarrow^+ y) \;.$$

$$\begin{aligned}\psi_1 \overset{\text{def}}{=} &\exists x(P_a(x) \wedge \mathsf{root}(x))\\ &\wedge \forall x(P_a(x) \Rightarrow \exists y(P_b(y) \wedge x \downarrow y))\\ &\wedge \forall y(P_b(y) \Rightarrow \mathsf{leaf}(y) \vee \exists x(P_a(x) \wedge y \downarrow x)) \;.\end{aligned}$$

[1] 2. Give a closed first-order formula ψ_2 enforcing that every branch starting from an a-labelled position contains a b-labelled position. *Hint: You can use the following first-order formula:*

$$\mathsf{branch}(x,y) \overset{def}{=} x \downarrow^* y \wedge \mathsf{leaf}(y) \;.$$

$$\psi_2 \overset{\text{def}}{=} \forall x \forall y\big((\mathsf{branch}(x,y) \wedge P_a(x)) \Rightarrow \exists z(x \downarrow^+ z \wedge z \downarrow^* y \wedge P_b(z))\big) \;.$$

3. Let $\Sigma = \{a, b, c\}$ and consider the formula

$$\begin{aligned}\psi \overset{\text{def}}{=} &\forall x \forall z\big((P_a(x) \wedge x \neq z \wedge \mathsf{branch}(x,z))\\ &\Rightarrow \exists y\big(x \downarrow^+ y \wedge y \downarrow^* z \wedge P_c(y) \wedge \forall z(x \downarrow^+ z \wedge z \downarrow^+ y \Rightarrow P_b(z))\big)\big) \;.\end{aligned}$$

[2] (a) Give an equivalent PDL node formula.

$$[\downarrow^*]\,\big(a \Rightarrow \big[(\downarrow; b?)^*; \downarrow; (a \vee (b \wedge \mathsf{leaf}))?\big]\bot\big)$$

[3] (b) Give a complete deterministic (bottom-up) finite hedge automaton for the set of models of ψ.

Let $Q \stackrel{\text{def}}{=} \{q_\perp, q_a, q_c\}$ and $Q_f \stackrel{\text{def}}{=} \{q_a, q_c\}$. The intuition is for

- $t \to^* q_\perp$ iff $t \not\models \psi$,
- $t \to^* q_a$ iff $t \models \psi$ and there is a branch with label in $b^* a \Sigma^* + b^+$, and
- $t \to^* q_c$ if $t \models \psi$ and every branch has a prefix in $b^* c$.

We use regular expressions over Q to describe the horizontal languages in the rules of Δ:

$$\begin{array}{rcc} a(q_c^*) \to q_a & a(Q^* \cdot (q_a + q_\perp) \cdot Q^*) \to q_\perp & \\ b(q_c^+) \to q_c & b(\varepsilon + (q_a + q_c)^* \cdot q_a \cdot (q_a + q_c)^*) \to q_a & b(Q^* \cdot q_\perp \cdot Q^*) \to q_\perp \\ c((q_a + q_c)^*) \to q_c & c(Q^* \cdot q_\perp \cdot Q^*) \to q_\perp & \end{array}$$

Based on: TD 5 Ex. 2

Exercise 2 (Propositional Dynamic Logic)**.** We work with unranked trees over a finite alphabet Σ.

1. We write $p \prec p'$ for two positions p and p' of a tree $t \in T(\Sigma)$ if p is visited before p' in a pre-order traversal of t. (Hence $\prec$ is a total order on $\mathrm{Pos}(t)$).

[1] Define a PDL path formula π such that $[\![\pi]\!]_t = \{(p, p') \in \mathrm{Pos}(t) \times \mathrm{Pos}(t) \mid p \prec p'\}$ for all $t \in T(\Sigma)$.

We start by defining a path formula for successors in a pre-order traversal, and then take its transitive closure:

$$\begin{aligned} \mathsf{succ} &\stackrel{\text{def}}{=} (\downarrow; \mathsf{first}?) + \big(\mathsf{leaf}?; (\mathsf{last}?; \uparrow)^*; \to\big) \\ \pi &\stackrel{\text{def}}{=} \mathsf{succ}^+ \end{aligned}$$

[2] 2. Define a PDL path formula π' such that $[\![(\pi')^*]\!]_t = \{(p, p') \in \mathrm{Pos}(t) \times \mathrm{Pos}(t) \mid t(p) = t(p')\}$ and $[\![\pi']\!]_t$ is a function for all $t \in T(\Sigma)$.

We build a new path formula on top of succ, which wraps around the root when we reach the rightmost leaf of t:

$$\begin{aligned} \mathsf{lastleaf} &\stackrel{\text{def}}{=} [\mathsf{succ}]\bot \\ \mathsf{wrap} &\stackrel{\text{def}}{=} \mathsf{succ} + (\mathsf{lastleaf}?; \uparrow^*; \mathsf{root}?) \\ \pi' &\stackrel{\text{def}}{=} \sum_{a \in \Sigma} a?; \mathsf{wrap}; (\neg a?; \mathsf{wrap})^*; a? \end{aligned}$$

Based on: TATA Ex. 1.6

Exercise 3 (Deterministic Top-Down Tree Automata)**.** Let t be a tree in $T(\mathcal{F})$ for some finite ranked alphabet $\mathcal{F}$ with maximal arity k, and let $\Pi \stackrel{\text{def}}{=} \left(\bigcup_{1\leq n\leq k} \mathcal{F}_n \times \{1,\dots,n\}\right)^* \cdot \mathcal{F}_0$. The *path language* $\text{Paths}(t) \subseteq \Pi$ is defined by

$$\begin{aligned}\text{Paths}(a) &\stackrel{\text{def}}{=} \{a\} && \text{if } a \in \mathcal{F}_0 \text{ is a constant,}\\ \text{Paths}(f(t_1,\dots,t_n)) &\stackrel{\text{def}}{=} \bigcup_{1\leq i\leq n} \{(f,i)\} \cdot \text{Paths}(t_i) && \text{if } f \in \mathcal{F}_n \text{ for some } 1 \leq n \leq k.\end{aligned}$$

We lift this to $\text{Paths}(L) \stackrel{\text{def}}{=} \bigcup_{t\in L} \text{Paths}(t)$ for any $L \subseteq T(\mathcal{F})$.

[4] 1. Show that if $L \subseteq T(\mathcal{F})$ is recognisable, then $\text{Paths}(L)$ is recognisable over the alphabet $\Sigma \stackrel{\text{def}}{=} \mathcal{F}_0 \cup \bigcup_{1\leq n\leq k} \mathcal{F}_n \times \{1,\dots,n\}$.
Hint: Start with a co-accessible top-down NFTA for L.

Let $\mathcal{A} = \langle Q, \mathcal{F}, Q_f, \Delta\rangle$ be a top-down NFTA with $L(\mathcal{A}) = L$. Without loss of generality, $\mathcal{A}$ is co-accessible: $\forall q \in Q$, $\exists t \in T(\mathcal{F})$, $q \to^*_{\mathcal{A}} t$.

We construct $\mathcal{A}' = \langle Q', \Sigma, \delta, I, F\rangle$ a NFA with state set $Q' \stackrel{\text{def}}{=} Q \uplus \{q_\ell\}$, initial state set $I \stackrel{\text{def}}{=} Q_f$, accepting state set $F \stackrel{\text{def}}{=} \{q_\ell\}$, and transition set

$$\begin{aligned}\delta \stackrel{\text{def}}{=} &\{(q,(f,i),q_i) \mid 0 < i \leq n \leq k,\ f \in \mathcal{F}_n,\ (q \to f(q_1,\dots,q_n)) \in \Delta\}\\ \cup\ &\{(q,a,q_\ell) \mid a \in \mathcal{F}_0,\ (q \to a) \in \Delta\}\ .\end{aligned}$$

We denote by $L_q(\mathcal{A}') \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid q \xrightarrow{w}_{\mathcal{A}'} q_\ell\}$ the word language recognised by $q \in Q$ in $\mathcal{A}'$; then $L(\mathcal{A}') = \bigcup_{q\in I} L_q(\mathcal{A}')$.

$\text{Paths}(L) \subseteq L(\mathcal{A}')$: We prove by induction on $t \in T(\mathcal{F})$ that, if $q \to^*_{\mathcal{A}} t$, then $\text{Paths}(t) \subseteq L_q(\mathcal{A}')$. Thus, if $t \in L$, then $q \in Q_f = I$, and $\text{Paths}(t) \subseteq L(\mathcal{A}')$.

- For the base case where $t = a \in \mathcal{F}_0$, $a \to q$ implies $(q, a, q_\ell) \in \delta$ and thus $a \in L_q(\mathcal{A}')$.
- For the induction step where $t = f(t_1,\dots,t_n)$, $f \in \mathcal{F}_n$ for some $1 \leq n \leq k$, we have the reduction $q \to_{\mathcal{A}} f(q_1,\dots,q_n) \to^*_{\mathcal{A}} t$ for some $f(q_1,\dots,q_n) \to q$ in Δ. We apply the induction hypothesis on each $q_i \to^*_{\mathcal{A}} t_i$ for $1 \leq i \leq n$:

$$\begin{aligned}\text{Paths}(f(t_1,\dots,t_n)) &= \bigcup_{1\leq i\leq n} \{(f,i)\} \cdot \text{Paths}(t_i) && \text{by def.}\\ &\subseteq \bigcup_{1\leq i\leq n} \{(f,i)\} \cdot L_{q_i}(\mathcal{A}') && \text{by ind. hyp.}\\ &\subseteq L_q(\mathcal{A}') && \forall 1 \leq i \leq n,\ (q,(f,i),q_i) \in \delta\ .\end{aligned}$$

$L(\mathcal{A}') \subseteq \text{Paths}(L)$: First note that $L(\mathcal{A}') \subseteq \Pi$. We show by induction on $w \in \Pi$ that, for all $q \in Q$, if $w \in L_q(\mathcal{A}')$, then there exists $t \in T(\mathcal{F})$ such that $w \in \text{Paths}(t)$ and $q \to^*_{\mathcal{A}} t$. Then, $w \in L(\mathcal{A}')$ occurs when $q \in I = Q_f$ and thus there exists $t \in L$ with $w \in \text{Paths}(t)$.

- For the base case where $w = a \in \mathcal{F}_0$, $w \in L_q(\mathcal{A}')$ requires $(q, a, q_\ell) \in \delta$, hence $t \stackrel{\text{def}}{=} a$ fits: $w \in \{a\} = \text{Paths}(t)$ and $q \to_{\mathcal{A}} a$.
- For the induction step, $w = (f, i)w_i$ for some $1 \leq i \leq n \leq k$, $f \in \mathcal{F}_n$, and $w_i \in \Pi$. Then there exists $q_i \in Q$ such that $(q, (f, i), q_i) \in \delta$ and $w_i \in L_{q_i}(\mathcal{A}')$; therefore there is a rule $q \to f(q_1, \ldots, q_i, \ldots, q_n)$ in Δ for some $q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_n \in Q$. By induction hypothesis, there exists $t_i \in T(\mathcal{F})$ such that $w_i \in \text{Paths}(t_i)$ and $q_i \to^*_{\mathcal{A}} t_i$. For all $j \in \{1, \ldots, n\} \setminus \{i\}$, as the state q_j is co-accessible, there exist a tree t_j such that $q_j \to^*_{\mathcal{A}} t_j$. Letting $t \stackrel{\text{def}}{=} f(t_1, \ldots, t_n)$, we have therefore $q \to^*_{\mathcal{A}} t$ and $(f, i) \cdot w_i \in \text{Paths}(t)$ as desired.

2. The *path closure* of a word language $L' \subseteq \Pi$ is

$$\overline{L'} \stackrel{\text{def}}{=} \{t \in T(\mathcal{F}) \mid \text{Paths}(t) \subseteq L'\} \; .$$

[3] Show that if $L' \subseteq \Pi$ is recognisable, then $\overline{L'} \subseteq T(\mathcal{F})$ is recognisable by a deterministic top-down tree automaton.

Let $\mathcal{A}' = \langle Q', \Sigma, \delta, I, F\rangle$ be a DFA recognising $L' \subseteq \Pi$. Observe that L' is *prefix*: if $ww' \in L'$ and $w \in L'$ for some $w, w' \in \Sigma^*$, then $w' = \varepsilon$; this is because the symbols of $\mathcal{F}_0$ act as end-of-word markers. Thus without loss of generality, F is a singleton $\{q_\ell\}$.

We construct a deterministic top-down tree automaton $\mathcal{A} = \langle Q, \mathcal{F}, Q_f, \Delta\rangle$ with $Q \stackrel{\text{def}}{=} Q' \setminus \{q_\ell\}$, $Q_f \stackrel{\text{def}}{=} I$, and

$$\begin{aligned}\Delta \stackrel{\text{def}}{=} &\{q \to a \mid a \in \mathcal{F}_0,\, \delta(q, a) = q_\ell\}\\ &\cup \{q \to f(\delta(q, (f, 1)), \ldots, \delta(q, (f, n))) \mid f \in \mathcal{F}_n \text{ and } \forall 1 \leq i \leq n, \delta(f, i) \text{ is defined}\} \; .\end{aligned}$$

We show by induction on $t \in T(\mathcal{F})$ that, for all $q \in Q$, $\text{Paths}(t) \subseteq L_q(\mathcal{A}')$, if and only if $q \to^*_{\mathcal{A}} t$. Then, $t \in \overline{L'}$, if and only if $\text{Paths}(t) \subseteq L'$, if and only if $\text{Paths}(t) \subseteq L_q(\mathcal{A}')$ for some $q \in I$, if and only if $q \to^*_{\mathcal{A}} t$ for some $q \in Q_f$, if and only if $t \in L(\mathcal{A})$.

- For the base case $t = a \in \mathcal{F}_0$, $\text{Paths}(a) = \{a\} \subseteq L_q(\mathcal{A}')$ if and only if $\delta(q, a) = q_\ell$, if and only if $(q \to a) \in \delta$ as desired.
- For the induction step, let $t = f(t_1, \ldots, t_n)$ for some $1 \leq n \leq k$, $f \in \mathcal{F}_n$, and each $t_i \in T(\mathcal{F})$. Then $\text{Paths}(t) = \bigcup_{1 \leq i \leq n}\{(f, i)\} \cdot \text{Paths}(t_i) \subseteq L_q(\mathcal{A}')$ if and only if $\text{Paths}(t_i) \subseteq L_{\delta(q,(f,i))}(\mathcal{A}')$ for all $1 \leq i \leq n$. By induction hypothesis, this is if and only if $\delta(q, (f, i)) \to^*_{\mathcal{A}} t_i$ for each i, which is if and only if $q \to_{\mathcal{A}} f(\delta(q, (f, 1)), \ldots, \delta(q, (f, n))) \to^*_{\mathcal{A}} f(t_1, \ldots, t_n) = t$ as desired.

[2] 3. Deduce that $L \subseteq T(\mathcal{F})$ is recognisable by a deterministic top-down tree automaton if and only if L is recognisable and *path closed*, i.e. $L = \overline{\text{Paths}(L)}$.

If L is recognisable by a deterministic top-down tree automaton $\mathcal{A}$, then L is recognisable and the automaton $\mathcal{A}'$ constructed in Question 1 for $L' \stackrel{\text{def}}{=} \text{Paths}(L)$ is a DFA. If we apply the construction of Question 2 to $\mathcal{A}'$ we obtain $\mathcal{A}$ back! Hence $L = L(\mathcal{A}) = \overline{\text{Paths}(L)}$.

Conversely, if L is recognisable and path closed, then by Question 1 $\text{Paths}(L)$ is recognised by a word automaton $\mathcal{A}'$, which we can determinise to obtain by Question 2 a deterministic top-down tree automaton for $\overline{\text{Paths}(L)} = L$.

[1] 4. Show that it is decidable whether a recognisable tree language is path closed.

This is clearly decidable since by questions 1 and 2 we can build a deterministic top-down tree $\mathcal{A}_d$ automaton of exponential size with $L(\mathcal{A}_d) = \overline{\text{Paths}(L)}$. As $L \subseteq \overline{\text{Paths}(L)}$ always holds, it suffices to check whether $L(\mathcal{A}_d) \subseteq L(\mathcal{A})$, i.e. whether $L(\mathcal{A}) \cap (T(\mathcal{F}) \setminus L(\mathcal{A}_d)) = \emptyset$. Observe that complementing $\mathcal{A}_d$ is trivial, hence this last inclusion test is in polynomial time in the size of $\mathcal{A}$ and $\mathcal{A}_d$, hence in EXP overall.

5. Let $\mathcal{F} \stackrel{\text{def}}{=} \{\wedge^{(2)}, \vee^{(2)}, \bot^{(0)}, \top^{(0)}\}$ and $L \stackrel{\text{def}}{=} \{t \in T(\mathcal{F}) \mid e(t) = \top\}$ be the set of trees that evaluate to $\top$ according to:

$$e(\wedge(t_1, t_2)) \stackrel{\text{def}}{=} e(t_1) \wedge e(t_2)\,, \quad e(\vee(t_1, t_2)) \stackrel{\text{def}}{=} e(t_1) \vee e(t_2)\,, \quad e(\bot) \stackrel{\text{def}}{=} \bot\,, \quad e(\top) \stackrel{\text{def}}{=} \top\,.$$

[1] Show that L is not recognised by any deterministic top-down tree automaton.

Indeed, $t_1 \stackrel{\text{def}}{=} \vee(\bot, \top)$ and $t_2 \stackrel{\text{def}}{=} \vee(\top, \bot)$ are in L. Thus $(\vee, 1)\bot \in \text{Paths}(t_1)$ and $(\vee, 2)\bot \in \text{Paths}(t_2)$ show that $t_3 \stackrel{\text{def}}{=} \vee(\bot, \bot) \in \overline{\text{Paths}(L)}$ although it does not belong to L.

[2] 6. Show that L is not recognised by any finite union of deterministic top-down tree automata.

Assume $L = \bigcup_{1 \leq i \leq n} L_i$ where each L_i is recognised by a deterministic top-down tree automaton. Consider the trees

$$t_0 \stackrel{\text{def}}{=} \vee(\top, \bot)\,, \qquad t_{m+1} \stackrel{\text{def}}{=} \vee(\bot, t_m)\,.$$

As all of these infinitely many trees belong to L, there must be $1 \leq i \leq n$ such that $t_j \in L_i$ and $t_k \in L_i$ for $j < k$. Hence $(\vee, 2)^{j-1}(\vee, 2)\bot \in \text{Paths}(t_j) \subseteq \text{Paths}(L_i)$ and $(\vee, 2)^{\ell}(\vee, 1)\bot \in \text{Paths}(t_k) \subseteq \text{Paths}(L_i)$ for all $\ell < j$ imply that $t'_j \in L_i = \overline{\text{Paths}(L_i)}$ where

$$t'_0 \stackrel{\text{def}}{=} \vee(\bot, \bot)\,, \qquad t'_{m+1} \stackrel{\text{def}}{=} \vee(\bot, t'_m)\,.$$

This contradicts $L_i \subseteq L$.